WHAT IS CLAIMED IS:

1. A method for processing divergent graphics samples in a programmable graphics processing unit, the method comprising:

incrementing a subroutine depth of a first sample to designate that first call instructions are to be executed on the first sample;

pushing state data of a second sample upon which the first call instructions are not to be executed onto a global stack; and

executing the first call instructions on the first sample.

2. The method of claim 1, further comprising the step of holding the second sample idle.

3. The method of claim 2, wherein holding the second sample idle comprises encoding the second sample with non-operation information.

4. The method of claim 1, further comprising the step of determining whether the first call instructions include a call return that contains second call instructions.

5. The method of claim 4, further comprising the step of incrementing the subroutine depth of the first sample to designate that second call instructions are to be executed on the first sample.

6. The method of claim 5, further comprising the step of executing the second call instructions on the first sample.

7. The method of claim 1, wherein pushing state data removes the second sample from a working set of data.

8. The method of claim 1, further comprising the step of determining whether an instruction in an instruction sequence includes a call-return that contains the first call instructions.

9. A method of processing divergent graphics samples in a programmable graphics processing unit, the method comprising:

identifying a first sample having a first subroutine depth;

holding idle a second sample having a second subroutine depth, the first subroutine depth being greater than the second subroutine depth;

executing first return instructions on the first sample; and

popping state data of a second sample from a global stack.

10. The method of claim 9, wherein holding idle a second sample comprises encoding the second sample with non-operation information.

11. The method of claim 9, wherein popping the state data of the second sample restores the second sample to a working set of data.

12. The method of claim 9, further comprising the step of decrementing the first subroutine depth, making the first subroutine depth equal to the second subroutine depth.

13. The method of claim 12, further comprising the step of determining whether the first subroutine depth is greater than zero.

14. The method of claim 13, further comprising the step of identifying in an instruction sequence a next instruction to be executed if the first subroutine depth is equal to zero.

15. The method of claim 13, further comprising the step of executing second return instructions on the first sample and the second sample if the first subroutine depth is greater than zero.

16. The method of claim 15, further comprising the step of decrementing the first subroutine depth and the second subroutine depth.

17. A system for processing divergent graphics samples in a programmable graphics processing unit, the system comprising:

a subroutine depth scoreboard configured to store a first subroutine depth corresponding to a first sample and a second subroutine depth corresponding to a second sample;

a global stack configured to store state data; and

a remap configured to increment and decrement the first subroutine depth and the second subroutine depth in the subroutine depth scoreboard and to push state data onto and to pop state data from the global stack.

18. The system of claim 17, wherein the remap is further configured to determine that first call instructions are to be executed on the first sample, but not the second sample, to increment the first subroutine depth to designate that the first call instructions are to be executed on the first sample, and to push state data of the second sample onto the global stack.

19. The system of claim 18, wherein the remap is further configured to encode the second sample with non-operation data, to generate a PC token for executing the first call instructions, the PC token containing one or more codewords that configure programmable computation units within a recirculating shader pipeline to execute the first call instructions, and to dispatch the PC token into the recirculating shader pipeline, followed by the first sample and the second sample.

20. The system of claim 17, wherein the remap is further configured to determine that first return instructions are to be executed on the first sample, but not the second sample, to encode the second sample with non-operation data, to generate a PC token for executing the first return instructions, the PC token containing one or more codewords that configure programmable computation units within a recirculating shader pipeline to execute the first return instructions, and to dispatch the PC token into the recirculating shader pipeline, followed by the first sample and the second sample.

21. The system of claim 20, wherein the remap is further configured to decrement the first subroutine depth and to pop state data of the second sample from the global stack.

22. A system for processing divergent graphics samples in a programmable graphics processing unit, the system comprising:

means for incrementing a first subroutine depth of a first sample to designate that first call instructions are to be executed on the first sample;

means for pushing state data of a second sample upon which the first call instructions are not to be executed onto a global stack;

means for executing the first call instructions on the first sample;

means for identifying that the first subroutine depth is greater than a second subroutine depth of the second sample;

means for executing first return instructions on the first sample;

means for decrementing the first subroutine depth; and

means for popping state data of the second sample from the global stack.

23. The system of claim 22, further comprising means for holding the second sample idle.

24. The system of claim 23, wherein means for holding comprises encoding the second sample with non-operation information.

25. The system of claim 22, further comprising means for determining whether the first subroutine depth is greater than zero and means for executing second return instructions on the first sample and the second sample if the first subroutine depth is greater than zero.